IRM Arcnet Support

Required for SRMs

Tue, Jan 11, 2000

This note describes the implementation of Arcnet support for communication with SRMs in the Linac control system. Each VME crate includes an Arcnet controller board, which typically allows access to five SRMs (Smart Rack Monitors), each of which is an Arcnet node. Each 15 Hz cycle, all available data is requested from each SRM and placed into the data pool for that Linac front end node.

The message protocol used on the Arcnet is not based upon IP, but rather is based upon the Acnet header in common use at the Fermilab Beams Division. This 18-byte header is followed by a short message format that supports request and settings. The document describing the detailed protocol used is called *SRM Message Protocols*.

The Acnet header support is provided by routines in the `NetLayer` module. Calling these routines to transmit a message results in a pointer to the message block being queued to the Arcnet Output Pointer Queue, or `ARCPQ`, according to a value of the destination node being in the range `0x7AA1—0x7ABF`, which permits up to 31 possible destination nodes, or `0x7A00`, which means broadcast to Arcnet. If we have five SRMs, they are usually numbered `0x7AA1—0x7AA5`, which correspond to the one-byte Arcnet node addresses `A1—A5`. It is when the `OUTPQX` routine is called to queue a message to a network that the pointer to the message block is written into `ARCPQ`.

At some point, the `NetSend` routine is called to flush all queued network messages to the network hardware. The logic of concatenating multiple consecutively-queued messages that target the same destination node is handled by the `NetXmit` code in `NetIntIP`. Finally, the routine is called that commits the assembled Arcnet frame to the Arcnet hardware, which is done via another queue called the Transmit Parameter List, which is borrowed from the scheme used by the Token Ring chipset. After being placed onto this queue, if the Arcnet hardware is currently *not* busy, an Arcnet interrupt is forced to "prime the pump" and get the frame sent to the controller. If the Arcnet hardware is already busy, then this step will automatically be performed once an Arcnet transmit interrupt is received and this queue entry is the next one to be passed to the controller. The objective here is to prevent holding up the building of additional Arcnet frames from message blocks that have been queued via `ARCPQ`. The Transmit Parameter List queue is large enough to hold references to 127 frames. The `ARCPQ` is large enough to hold pointers to 1K messages. And the transmit frame circular buffer is 48K bytes in length. The maximum length of an Arcnet frame is 508 bytes. These limits probably represent overkill. The typical size of a transmitted frame containing one message is less than 30 bytes.

For frame reception, the Arcnet receive interrupt copies the frame from the hardware input buffer into a circular buffer area used for frame reception, of size 64K bytes. In practice, most received frames are of size 256 bytes or less. But five may be received in each 15 Hz cycle. Still, the nature of 15Hz operation is such that we must keep up with all such frames, so 64K is enough.

Client support for Arcnet is provided by `SRMReq` and `SRMSet` routines. The `SRMReq` routine is only called by the Data Access Table handler for the type `0x20` entry. This sends a request to an `SRM`, which usually is broadcast to all `SRM`s on the Arcnet, asking for each to read all its data and return it in response. Other possible requests are to ask for memory data, or to ask for a given table# contents. Only the standard Memory Dump page uses the format for memory access. It must emulate the functionality of `SRMReq` for the memory address case. The request format for a table# contents is not

used at present.

Various small helper routines invoke `SRMSet` to build a setting message and queue it to the network via `OUTPQX`. The `SRMAC` routine is used for analog control, `SRMMC` is used for motor control, SRMDC is used for digital bit control, and `SRMDCB` and `SRMDCW` are use for digital byte and word control. All of these routines result in USM-type logic in performing the setting; no setting acknowledgment is requested.

In both `SRMReq` and `SRMSet`, the `SRMQueue` routine is called, followed by `FLUSHNET`, which causes an event to be sent to the `Update` Task that will result in `NetSend` being called to flush all queued messages to the network. Note that this scheme permits multiple messages, say settings, to be queued up and sent in a single Arcnet frame to the SRM. This would of course happen during Settings Restore following system initialization, but it could also happen as a result of multiple settings received from a client.